

# 案例 2-旋转的金刚石图案

档编写：霍波魏

校稿/修订：孔令德

时间 2019~2020

联系方式：QQ997796978

**说明：**本套案例由孔令德开发，原版本为 Visual C++6.0，配套于孔令德的著作《计算机图形学-基于 MFC 三维图形开发》一书。孔令德计算机工程研究所的学生霍波魏在学习计算机图形学期间，对本套案例进行了升级并编写了学习文档。现在程序的编写和程序的解释都是基于 Windows 10 操作系统，使用 Microsoft visual studio 2017 平台的 MFC（英文版）开发。

## 一、知识点

### 1、双缓冲绘图

一次性将图形绘制到内存缓冲区，然后使用源拷贝方式将图形拷贝到设备缓冲区。

### 2、定时器

为程序设置定时器，每隔一定时间间隔调用双缓冲绘图，绘图时改变金刚石的转角。

## 二、案例描述

以屏幕坐标系原点为圆心，绘制了一个 19 个顶点的金刚石图案。在定时器的作用下，让其绕旋转原点逆时针旋转。设置工具栏图标按钮来启动或者停止金刚石的旋转。

## 三、实现步骤

1. 添加绘制函数 DrawObject()，对金刚石进行绘制。
2. 添加 DoubleBuffer()双缓冲函数，并对 DrawObject()进行调用。
3. 在 OnDraw()函数中对双缓冲进行调用。
4. 状态栏的修改以及对 bPlay 进行设置。

## 四、主要算法

### 1. CTestView 类

```
void CTestView::DoubleBuffer(CDC* pDC) //双缓冲
{
```

```

CRect rect;//定义客户区矩形
GetClientRect(&rect);//获得客户区的大小
pDC->SetMapMode(MM_ANISOTROPIC);//pDC 自定义坐标系
pDC->SetWindowExt(rect.Width(), rect.Height());//设置窗口范围
pDC->SetViewportExt(rect.Width(), -rect.Height());//设置视区范围, x 轴水平向右, y
轴垂直向上
pDC->SetViewportOrg(rect.Width() / 2, rect.Height() / 2);//客户区中心为原点
CDC memDC;//内存 DC
CBitmap NewBitmap, *pOldBitmap;//内存中承载的临时位图
memDC.CreateCompatibleDC(pDC);//创建一个与显示 pDC 兼容的内存 memDC
NewBitmap.CreateCompatibleBitmap(pDC, rect.Width(), rect.Height());//创建兼容位
图
pOldBitmap = memDC.SelectObject(&NewBitmap);//将兼容位图选入 memDC
memDC.SetMapMode(MM_ANISOTROPIC);//memDC 自定义坐标系
memDC.SetWindowExt(rect.Width(), rect.Height());
memDC.SetViewportExt(rect.Width(), -rect.Height());
memDC.SetViewportOrg(rect.Width() / 2, rect.Height() / 2);
rect.OffsetRect(-rect.Width() / 2, -rect.Height() / 2);
Object->DrawObject(&memDC);//向 memDC 绘制图形
pDC->BitBlt(rect.left, rect.top, rect.Width(), rect.Height(), &memDC,
-rect.Width() / 2, -rect.Height() / 2, SRCCOPY);//将内存 memDC 中的位图拷贝到显示
pDC 中
memDC.SelectObject(pOldBitmap);//恢复位图
NewBitmap.DeleteObject();//删除位图
}
void CTestView::DrawObject(CDC* pDC) //绘制金刚石图案函数
{
    CPen NewPen, *pOldPen;//定义画笔
    NewPen.CreatePen(PS_SOLID, 1, RGB(255, 255, 255));//创建画笔
    pOldPen = pDC->SelectObject(&NewPen); //将蓝色画笔选入设备上下文
    double Theta; //定义金刚石图案的起始角与等分角
    Theta = 2 * PI / nCount; //θ 为等分角
    for (int i = 0; i < nCount; i++) //计算等分点
    {
        P[i].x = ROUND(nRadius * cos(i * Theta + Alpha));
        P[i].y = ROUND(nRadius * sin(i * Theta + Alpha));
    }
    for (int i = 0; i <= nCount - 2; i++)//连接各点
    {
        for (int j = i + 1; j <= nCount - 1; j++)
        {
            pDC->MoveTo(nWidth + P[i].x, nHeight - P[i].y);
            pDC->LineTo(nWidth + P[j].x, nHeight - P[j].y);
        }
    }
}

```

```

    }
    pDC->SelectObject(pOldPen); //恢复设备上下文中的原画笔
    NewPen.DeleteObject(); //删除已成自由状态的蓝色画笔
}

```

## 五、实现效果

金刚石图案效果图如图 2-1。

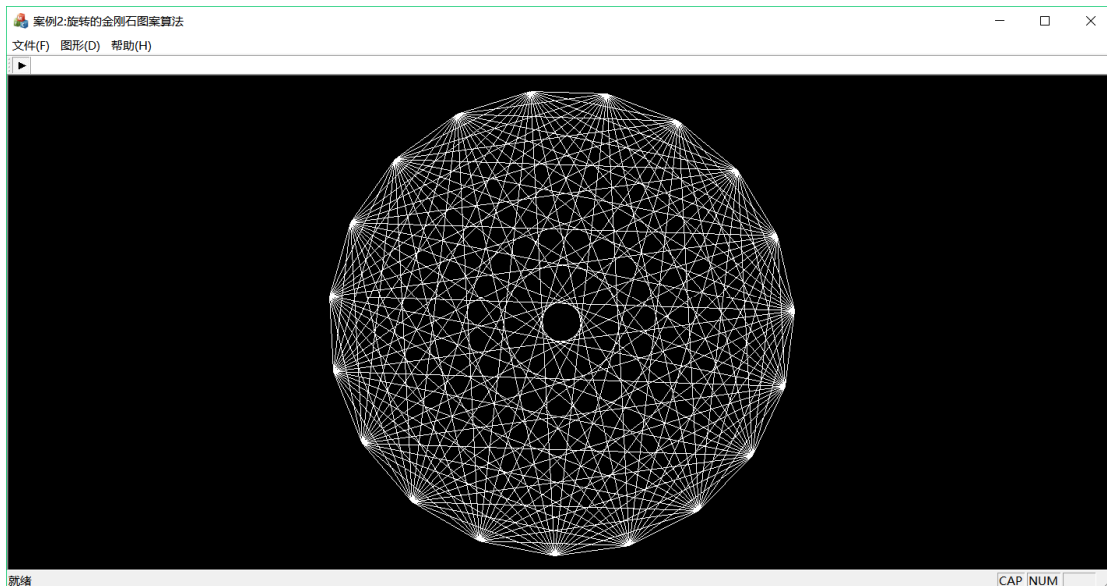


图 2-1 金刚石效果图

## 六、遇到的问题及解决方案

### 1. 添加启动图标按钮

通过设置菜单命令和 Command 命令来控制。

### 2. 改变旋转角度

在 WM\_TIMER 消息映射函数中改变旋转角度。

### 3. 擦除背景

通过 WM\_ERASEBKGDND 消息映射函数在窗口大小发生改变等情况下发生的，它将绘制窗口背景。

### 4. C++ 中 \_T()函数和 \_L()函数

“\_T(”是一个宏,他的作用是让你的程序支持 Unicode 编码。“L”是表示字符串资源为 Unicode 的,“\_T”是一个适配的宏,有#ifdef \_UNICODE 的时候“\_T”就是“L”,没有#ifdef \_UNICODE 的时候“\_T”就是 ANSI 的,“\_T”和“L”的区别在于“L”不管你是以什么方式编译,一律以 UNICODE

方式保存。

## 七、案例总结

本案例采用的 19 个等分点绘制的，图形在中心是一个圈，但是当等分点变成 20 个时，所有点的连线就汇聚到了一个点上，图形的中心就汇聚到一个点；由此可以推广到等分点为奇数个时，图形的中心是一个圈，等分点为偶数时，图形的中心为一个点。

本案例的金刚石的算法较为简单，但发现自己的写程序的思路还不是很清晰，导致在编译的时候总是会有忘了调用或者忘了初始化的情况出现。